428 Rec'd PCT/PTO 25 JAN 2000
09/463527

Ins. A1

~~SPECIFICATION~~

~~METHOD AND SWITCHING UNIT FOR COMMUNICATING DATA ACCORDING TO THE ATM PROTOCOL AND THE INTERNET PROTOCOL~~

The present invention is directed to a method for operating a communication network, whereby data frames defined according to a first protocol are employed, these also containing a destination address in addition to the ~~payload~~ message data to be transmitted, said destination address defining the receiver of the data frame. For transmission, data packets defined according to a second protocol are generated from the data of a data frame, these also containing a connection identifier in addition to the data of the data frame that defines the receiver of the data packet. In the receiver of the data packets of a data frame, the destination address is read from the data packet containing the destination address. A new connection identifier is then determined on the basis of the destination address, this determining a new receiver of the data packets. Subsequently, new data packets that contain the new connection identifier are generated from the received data packets of the data frame. The data of the data frame ~~are~~ is also checked for transmission errors according to a predetermined error checking method, whereby reference data in the data frame ~~contain~~ contains a rated value for the error checking. The new data packets of a data frame received error-free are sent to the new receiver.

The article "a^{I}t^{P}m: Strategy For Integrating IP With ATM", by G. Parulkar, D. C. Schmidt and J. S. Turner in SIGCOMM '95, Cambridge, MA USA, page 49 through page 58, explains a method for operating a communication network wherein the Internet protocol, referred to in brief (IP) is utilized as first protocol and the ATM protocol (asynchronous transfer mode) is utilized as second protocol. The article is also particularly directed to the construction of what is referred to as a router with which the data frames or~~, respectively,~~ the data packets are further-communicated to their receiver dependent on the traffic load. Section 3.1 of said article explains

that, in a mode for short messages, each data frame (IP packet) is processed with the assistance of a software. The validity of a header field of the data frame is thereby checked. A routing decision is made and data in the header field of the data frame ~~are~~ is modified. Before the data packets of the data frame are processed with the assistance of the software, however, all data packets of a data frame are stored in a memory. What is disadvantageous about this method is that the processing of the data packets is delayed by this intermediate storing, so that a delay also occurs in the transmission of the data packets and, thus, of the data frame as well. This delay is all the more serious the higher the ~~plurality~~ number of incoming data packets per time unit and the more routers that are employed in the transmission, so that the delay times sum up.

For solving this problem, a mode for longer messages is explained in Section 3.2 of said article wherein only the first data frame is processed as demanded in the Internet protocol. All other data frames of the message are forwarded via a dialed connection and not processed with the assistance of the software. What is disadvantageous about this method in addition to the infringement of the Internet protocol is that additional method steps are required for the setup of the dialed connection. Moreover, circuit-oriented measures must be undertaken that support this type of data transmission.

SUMMARY OF THE INVENTION

~~An~~ It is an object of the present invention ~~is to specify~~ to provide a simple method for operating a communication network wherein a ~~plurality~~ number of protocols are to be adhered to in the switching of the data packets.

Ins. a2

~~This object is achieved by a method having the features of patent claim 1. Advantageous developments are recited in the subclaims.~~

The present invention proceeds from the consideration that it is usually possible to already generate new data packets from the data packets already received for a data frame before the reception of the last data packet of this data frame. Whether the received data packets have been correctly transmitted, however, can only be determined with the reception of the last data packet of a frame given the protocols usually employed. The present invention accepts that the generation of the new data packets is possibly unnecessary

because transmission errors occurred in the transmission of the received data packets of the data frame, these no longer being capable of being corrected. In this case, the received data packets and the new data packets as well are to be discarded. This disadvantage, however, can be accepted since no additional circuit-oriented measures need be undertaken and the circuit unit already provided would otherwise be unused until the reception of the last data packet of the data frame.

In ~~the method~~ an embodiment of the invention, the generation of the new data packets begins before all data packets of the data frame have been received. What this measure achieves is that the transmission of the new data packets can already be ~~begun~~ started shortly after the reception of the last data packet of the data frame since only a part of the new data packets, for example only the last new data packet, must still be generated. The delay is only influenced by the time for checking the result of the error checking method. This time is usually very short since only a simple comparison of a calculated error value to a rated value usually stored in the last packet of a data frame need ensue. The comparison result is thus available in a circuit arrangement as soon as the error value is calculated and as soon as the last data packet has been received. The last new data packet of the data frame is preferably generated in ~~the method of~~ the invention while the new data packets of the data frame that have already been generated are being sent. The short delay time that occurs - particularly given a high data throughput, i.e. a great number of data ~~are~~ is forwarded per second - leads thereto that data packets have to be rejected only very seldom. The ~~method of the~~ invention can thus be advantageously utilized particularly when the data rates lie in ranges of 50 megabits per second or even far above this value.

In ~~a development of the method~~ an embodiment of the invention, at least one datum of the data frame is modified. In this case, new reference data ~~are~~ is generated in agreement with the error-checking method for the data frame that ~~are~~ is employed instead of the previous reference data. As a result of this measure, the communication of the data frame ensues in agreement with the first protocol that defines the error-checking method. During further

processing, ~~and~~ an infringement of the protocol would increase the probability that errors are not recognized.

In ~~another development~~ an embodiment of the invention, the error checking method and/or the calculation of the new reference data is implemented keeping step with the generation of the new data packets of the data frame. As a result of this measure, the new data packets have to be processed only once. The ~~plurality~~ number of memory accesses can thus be reduced to a necessary, minimum extent. When the generation of the new data packets also ensues keeping step with the reception of the data packets, then the new data packets are nearly completely present when the last data packet of a data frame is received. The processing of the entire data frame thus ensues nearly simultaneously with the reception of the data frame. What keeping step thereby means is that the generation of a new data packet is implemented simultaneously with the reception of further data packets of the data frame. This assumes that the generation of a data packet is implemented in a time that is shorter ~~then~~ than the time for the reception of a data packet.

When, as prescribed in the Internet protocol, the counter value in every data frame that defines the "life span" remaining for this data frame is reduced in the method of the invention, then the data frames are switched in conformity with the protocol. The reduction of the counter value can assure that each data frame traverses only the number of transmission lengths prescribed by a start value of the counter. An overload of the network due to data packets that cannot be delivered to their receiver is prevented when data packets having the counter value 0 are not further-switched.

In ~~a development~~ an embodiment of the invention, a data packet of the data frame containing the destination address is recognized with the assistance of a first revaluation memory on the basis of its connection identifier. New connection identifiers are allocated to the incoming data packets with the assistance of the first revaluation memory. When the new connection identifier for the first data packet of a data frame has not yet been determined, then an entry is stored in the first revaluation memory that indicates that the new connection

identifier still has to be determined. When the new connection identifier ~~was~~ is determined, then the entry in the first revaluation memory is overwritten by the new connection identifier. The new connection identifier is then read from the first revaluation memory for all further data packets of the data frame. After the reception of the last data packet of a data frame, the entry is again stored in the revaluation memory for the connection identifier of the received data packets. As a result of this measure, a simple and fast possibility derives of determining the new connection identifiers for the data packets. The data packets are then further-switched on the basis of the new connection identifiers.

In an embodiment ~~another development~~ of the invention, a second revaluation memory is employed in order to allocate new connection identifiers to the destination addresses. The first-time determination of the new connection identifier ensues simply and quickly as a result of this measure. The second revaluation memory has the advantage that the stored, new connection identifiers can be inherently adapted to changing conditions in the communication network. A control unit overwrites the previously valid, new connection identifiers, with what are now valid, new connection identifiers in this case.

When no new connection identifier is stored in the second revaluation memory for the destination address of an incoming data frame, then the control unit will determine the connection identifier for this potentially new destination address and write it into the second revaluation memory. When no free memory is present in the second revaluation memory, the control unit can thereby also overwrite an already existing entry, for example that entry that remained unused for the longest time.

In this relatively rare case, that no new connection identifier is stored in the second revaluation memory for a destination address, the data frame must be stored until the control unit has determined the new connection identifier for the destination address of the data frame.

As a rule, the control unit has a large, non-volatile memory available to it in which a few 10,000 destination addresses can be stored. When it is

a matter of a new destination address that is not yet known in this control unit, the control unit - with the assistance of known methods - will interrogate other control units of the network in order to thus assign the correct, new ~~correction~~ connection identifier for this new destination address.

The revaluation memories are preferably associative memories, which are also referred to as "content addressable memory" in English. These memories have a very short access time. Moreover, the control unit externally accessing the associative memory need not implement a search procedure itself. This search procedure is already implemented in the associative memory without making use of the external control unit.

For switching data, the present invention is also directed to a switching unit that, in particular, is employed for the implementation of the method of the invention. The aforementioned technical effects also apply to the switching unit.

In ~~a development~~ an embodiment of the inventive switching unit, this also contains the two revaluation memories. When the revaluation memories are associative memories, then the switching of the data packets of a data frame ensues exclusively with one circuit arrangement. A software that works relatively slowly is no longer required. Only extremely slight delays arise in the switching unit when switching the data packets. As a result thereof, a multitude of data packets can be communicated per second, these, for example, containing a data quantity of 50 megabits or even far above this value.

Sub. a3

~~Exemplary embodiments of the invention are explained below with reference to the attached drawings. Shown therein are:~~

Figure 1 function units of a data communication network that works according to the Internet protocol and the ATM protocol;

Figure 2 the division of the data of what is referred to as a AAL5 onto ATM data packets;

~~Figure 3 the structure of a line unit;~~

Figure 4 an example of data transmission according to Internet protocol and ATM protocol;

Figure 5 entries in an IP revaluation memory and in an ATM revaluation memory given the data transmission according to Figure 4;

Figures 6 a and 6b a flow chart with method steps that are implemented in the switching.

Figure 1 shows electronic function units of a data communication network - referred to in brief as network 10 - that contains Internet computers 12, 14, 16 ~~12 through 16~~ that can send and receive data according to Internet protocol (IP). The Internet computers 12, 14, 16 ~~12 through 16~~ are also referred to as host computers.

The Internet computer 12 is connected via a transmission line 18 to a line unit 20 of a switching unit 22. In an IP-ATM interface of the Internet computer 12, ATM data packets - whose structure is explained with reference to Figure 2 - are generated from IP data packets to be sent - which are also explained below with reference to Figure 2. ATM data packets that the computer 12 receives, on the other hand, are converted into IP data packets in an ATM-IP interface. The structure of the line unit 20 as well as of further line units 28, 32 and 36 is explained in greater detail later with reference to Figure 3.

The switching unit 22 is also referred to as a "router" ~~in English~~. It has the job of switching data in the network 10 between sub-networks, one sub-network 24 thereof being shown in Figure 1. The Internet computers 12 through 16 are directly connected to the switching unit 22. The switching unit 22 reacts to errors in the network 10 and to modified load conditions in the network 10 in that the data packets it receives are alternatively routed. The switching unit 22 thereby attempts to find the respectively best path from one to another sub-network for the data packets, for example to the sub-network 24.

The Internet computer 14 is connected via a transmission line 26 to a line unit 28. Likewise, the Internet computer 16 is connected via a

transmission line 30 to a line unit 32. The line units 28 and 32 are component parts of the switching unit 22 and have the same structure as the line unit 20. The connection between the switching unit 22 and the sub-network 24 is produced via a transmission line 34 and a line unit 36. Two further Internet computers 38 and 40 are connected to the sub-network 24.

The switching unit 22 also contains a controller 42 and an ATM switching network 44. The controller 44 controls, among other things, the switching events in the switching network 44. The switching network 44, dependent on the control signals of the controller 42, can set up connections between the line units 20, 32, and 36. For example, a switching unit EWSX of Siemens AG is utilized as switching network 44. In an EWSX, what are referred to as line cards are usually arranged at the inputs of the switching network. These line cards are now replaced at some or all terminals of the switching network 44 by line units 20, 28, 32, 36 that, in addition to the functions of the line cards, assume the functions explained below such as, for example, what is referred as a traffic routing.

Figure 2 shows the division of the data of ~~a~~ an AAL5 data frame 50 (ATM Adaption Layer) onto ATM data packets 52 through 58. The AAL5 data frame 50 contains data of an IP data packet 60 that was generated by one of the Internet computers 12, 14, 16 ~~through 16~~ or~~, respectively~~, 38, 40, see Figure 1. This division has been defined in detail in the guideline RFC 1766 (request for comment) by what is referred to as the Internet Engineering Task Force, abbreviated as IETF, that is similar to a standard. The IP data packet 60 contains data that is ~~are~~ arranged according to Internet protocol, for which reason the data packet 60 is also referred to as IP data packet (Internet protocol data packet). The IP data packet 60 has a header part ~~60~~ 62 in which data for the implementation of the transmission is ~~are~~ stored, for example, a destination address that indicates the actual receiver of the IP data packet 60. The further structure of the header part 60 is explained later with reference to Figure 5. The IP data packet 60 also contains a ~~payload~~ message part 64 in which the ~~payload~~ message to be transmitted is ~~are~~ contained, for example data from a data file or data of an electronic letter (mail). The Internet

protocol merely prescribes a maximum length for the ~~payload~~ message part, which dare not be exceeded. The length of the ~~payload~~ message part 64 is thus variable and is noted in the header part 62 for the respective IP data packet 60.

The AAL5 data frame 50 contains the header part 62 and the ~~payload~~ message part 64 of the IP data packet 60. According to the guideline RFC 1766, the AAL5 data frame 50 has a length that is a multiple of 48 octets or, ~~respectively,~~ of 48 bytes. Filler data 66 ~~are~~ is inserted in order to meet this demand. Moreover, the data frame 50 contains reference data 68 that ~~contain~~ contains a rated value for a predetermined error checking method. This error checking method is, for example, a cyclical coding that is also referred to ~~in English~~ as "cyclic redundancy coding". A predetermined generator polynomial with which the reference data 68 ~~are~~ is generated is employed for the error checking method.

The AAL5 data frame 50 is subsequently respectively divided into sections having a length of 48 octets. These sections respectively form the ~~payload~~ message data in ATM data packets 52, 54, 56 or~~, respectively,~~ 58. Each ATM data packet additionally contains a header part composed of 5 octets wherein data for the implementation of the communication of the ATM data packets 52 through 58 ~~are~~ is stored, for example a connection identifier that defines the receiver of the respective data packet 52 through 58.

Figure 3 shows the structure of the line unit 20 that contains a processing unit 100, an IP revaluation memory 102, an ATM revaluation memory 104, a memory 106 and a memory 108. The processing unit 100 accepts the ATM data packets transmitted on the transmission line 18 and processes them according to the method explained below with reference to Figures 6a and 6b. The IP revaluation memory 102 is thereby employed, the Internet addresses IP-ADR, on the one hand, and, on the other hand, a connection identifier VPI/VCI for data packets (virtual path identifier/virtual channel identifier) belonging to each Internet address IP-ADR being stored therein. With the assistance of the IP revaluation memory 102, the processing unit 100 can determine which connection identifier VPI/VCI belongs at the moment to a specific Internet address-ADR. The connection

identifiers VPI/VCI in the revaluation memory 102 are updated by the controller 42 of Figure 1 dependent on the momentary transmission conditions in the network 10, see Figure 1.

Connection identifiers VPI/VCI for incoming or~~, respectively,~~ received ATM data packets and connection identifiers VPI/VCI for ATM data packets to be sent are stored in the ATM revaluation memory 104. The processing unit 100 transmits connection identifiers VCI/VPI read from the IP revaluation memory 102 for ATM data packets to be sent into the ATM revaluation memory 104. With the assistance of the revaluation memory 104, the processing unit 100 for connection identifiers in incoming ATM data packets can determine the connection identifiers for the ATM data packets to be sent. The ATM data packets to be sent, which are also referred to below as new data packets, are stored in the memory 106 for all IP data frames 50 processed at the moment - see Figure 2 - until all new ATM data packets of the IP data frame 50 have been generated by the processing unit 100.

During the generation of the new ATM data packets, a predetermined error checking method is implemented, intermediate values CRC and CRC* that are stored in the memory 108 being calculated upon implementation thereof.

When the processing unit 100 has ended the processing of the incoming data packets for an IP data frame 50, then the new ATM data packets for this IP data frame 50 stored in the memory 106 are transmitted via a transmission line 110 to the ATM switching network 44 and are communicated to a transmission line 112 by the latter on the basis of their connection identifier.

Figure 4 shows an example of an integrated data transmission according to Internet protocol and ATM protocol. Data ~~are~~ is thereby transmitted from the Internet computer 12 to the Internet computer 16 and from the Internet computer 12 to an Internet computer 120 that is connected to the switching unit 22 via a transmission line 122. The transmission line 122 ends at a line unit 124 in the switching unit 22, this line unit 124 being constructed like the line unit 20. The Internet computer 12 is also referred

to below as end system A and has an Internet address IP-ADR = 149.20.28.15. The ATM data packets sent by the Internet computer 12 have connection identifiers VPI_A/VCI_A, whereby the index A indicates a connection proceeding from the end system A. The Internet computer 16 is referred to below as end system B. It has the Internet address IP-ADR = 218.20.27.73. The ATM connection on the transmission line 30 to the end system B has the connection identifier VPI^*_B/VCI^*_B, whereby the index B again identifies the end system B. The asterisk "*" indicates that this connection identifier can be potentially revaluated again by the line unit 32, which, however, is not critical to the invention. The Internet computer 120 is referred to below as end system C and has the Internet address IP-ADR = 218.20.27.74. For the ATM connection to the end system C, a connection identifier VPI^*_C/VCI^*_C is employed on the transmission line 122.

In the transmission of data from the end system A to the end system B, the controller 42 connects what is referred to as a virtual connection 126 having a transmission identifier VPI_B/VCI_B in the ATM switching network 44. For the transmission of the data from the end system A to the end system C, subsequently, the controller 42 switches a virtual connection 128 having a transmission identifier VPI_C/VCI_C in the ATM switching network 44. The virtual connection 126 connects the line unit 20 to the line unit 32. The virtual connection 128, by contrast, connects the line unit 20 to the line unit 124.

During the further explanation of Figure 4, Figure 5 is also referenced, a part a thereof showing entries in the ATM revaluation memory 104 and in the IP revaluation memory 102 as well as an IP data 60' that is sent from the end system A to the end system B. The IP data packet 60' contains a data field 150 in which the version of the currently employed Internet protocol is noted, the arrangement of the data in the IP data packet 60' corresponding thereto. The processing priority for the IP data packet 60' is indicated in a data field 152. Control data ~~are~~ is contained in a data field 154. Among other things, the length of the IP data packet 60' as well as a counter value that indicates the ~~plurality~~ number of switchings of the IP data packet 60' that have

already ensued are stored in the data block 156. The Internet address of the end system B is recited as destination address in an address field 158. The Internet address of the end system A is indicated as source address in an address field 160. The data fields 150 through 154, the data block 156 as well as the address fields 158, 160 form the header part 62' of the IP data packet 60'. The ~~payload data are~~ message is located in a ~~payload~~ message part 64' of the IP data packet 60'.

As already mentioned, the IP data packet 60' is divided onto ATM data packets before the transmission. The data field 158 with the destination address is thereby located in the first ATM data packet that belongs to the IP data packet 60'. When the first ATM data packet of the IP data packet 60' is received by the line unit 20, then the processing unit 100 - see Figure 3 - reads the ATM revaluation memory 104. The ATM revaluation memory 104 usually contains an entry "IP" in the memory cells 162 and 164 at this time for the connection identifier VPI_A/VCI_A of the received ATM data packet with the destination address. When the connection identifier VPI_A/VCI_A is input in the ATM revaluation memory 104, then the entry "IP" is associatively output directly at the output of the ATM revaluation memory 104. On the basis of this entry, the processing unit 100 recognizes that a new connection identifier must be calculated for the ATM data packets of the IP data packets 60'.

The new connection identifier is stored in the IP revaluation memory 102. The IP revaluation memory 102 is likewise an associative memory. The destination address of the end system B is stored in a memory cell 166. The connection identifier VPI_B/VCI_B belonging to this destination address is stored in another memory cell 168 of the IP revaluation memory 102. When the destination address of the end system B is input into the IP revaluation memory, then the connection identifier VPI_B/VCI_B is associatively output at the output of the IP revaluation memory 102. The new connection identifier VPI_B/VCI_B for all ATM data packets of the IP data packet 60' is thus determined.

So that the new connection identifier VPI_B/VCI_B need not be determined anew by the processing unit 100 for each ATM packet of the IP data packet 60', the processing unit 100 enters the new connection identifier VPI_B/VCI_B ~~[sic]~~ into the memory cell 164 of the ATM revaluation memory 104. The new connection identifier ~~VPIB~~ VPI_B/VCI_B ~~[sic]~~ is directly read out from the ATM revaluation memory 104 for the other ATM data packets without accessing the IP revaluation memory in the meantime.

On the basis of the length of the IP data packet 60' stored in the data block 156, the processing unit 100 either determines how many ATM data packets belong to the IP data packet 60' or, on the basis of what is referred to as the PT (payload type) field of the header part of an ATM data packet (PTi = 001), recognizes the last ATM data packet of the IP data packet. After reception of all ATM data packets of the IP data packet 60', the entry "IP" is again entered into the memory cell 164 of the ATM revaluation memory 104. During reception of the ATM data packets of the IP data packet 60', ATM data packets having a different connection identifier can be sent to the line unit 20 from the end system A. The processing of these ATM data packets ensues analogous to the processing of the ATM data packets of the IP data packet 60'.

~~A part~~ Part b of Figure 5 shows the entries in the IP revaluation memory 102 and in the ATM revaluation memory 104 given the transmission of an IP data packet 60" following the transmission of the IP data packet 60'. As already mentioned, the IP data packet 60" is to be transmitted from the end system A to the end system C. The IP data packet 60" is also constructed according to Internet protocol. A header part 62" is constructed like the header part 62' of the IP data packet 60', so that its structure need not be explained again; the reference characters of the header part 62", however, are identified by two primes. The IP data packet 60" contains a ~~payload~~ message part 64" whose length deviates from that of the ~~payload~~ message part 64'. Accordingly, a different length is indicated in the data block 156". Moreover, the address of the end system C is stored in the address field 158" as destination address.

After the reception of the first ATM data packet of the IP data packet 60" for the connection identifier VPI_A/VCI_A, the entry "IP" is read from the ATM revaluation memory 104 in the line unit 120. This ATM data packet must thus be the first ATM data packet of the IP data packet 60". As already carried out in the transmission of the IP data packet 60', a new connection identifier VPI_C/VCI_C for the destination address of the end system C is determined on the basis of the entry "IP" with the assistance of the IP revaluation memory 102. This destination address is stored in a memory cell 170 of the IP revaluation memory 102. The connection identifier VPI_C/VCI_C ~~[sic]~~ belonging to this destination address is stored in a memory cell 172 of the IP revaluation memory 102. The new connection identifier VPI_C/VCI_C is subsequently stored in the memory cell 164 of the ATM revaluation memory 104, so that the IP revaluation memory 102 no longer need be accessed in the transmission of the other ATM data packets of the IP data packet 60". The switching can ensue exclusively with the assistance of the ATM revaluation memory 104. After the communication of all ATM data packets of the IP data packet 60", the entry "IP" is again entered into the memory cell 164.

As already mentioned, the IP revaluation memory 102 is updated by the controller, for example dependent on the traffic conditions in the network 10 (see Figure 1), in that potentially different connection identifiers are entered into the memory cells 168 and 172.

Figures 6A and 6B show a flow chart with method steps that are carried out in the line unit 20 given the transmission of ATM data packets. When explaining Figures 6A and 6B, Figures 3 through 5 are also referenced without expressed indication thereof. The method begins in a Step 200 with the activation of the switching unit 22. In a Step 202, a check is carried out with the assistance of the IP revaluation memory 102 to see whether the entry "IP" belongs to the connection identifier of the received ATM packet and, thus, the ATM data packet received at the moment is the first ATM data packet of an IP data packet. Another possibility is the recognition of the first ATM data packet of an IP data packet on the basis of the fact that this is the

first ATM data packet having the payload type PTi = 000 that follows an ATM data packet having the same connection identifier but the payload type PTi = 001. When this is not the case, then a standard processing wherein a new ATM data packet is generated from the received ATM data packet ensues in a Step 204, this new ATM data packet then being forwarded. In the standard processing, the new connection identifier is determined with the assistance of the ATM revaluation memory 104. The IP revaluation memory, by contrast, is not employed in the standard processing. Step 204 is again followed by Step 202, so that the method is in a loop composed of method Steps 202 and 204. The loop is only left in Step 202 when an ATM data packet processed at the moment is the first ATM data packet of an IP data packet. When this is the case, then a Step 205 follows immediately after the Step 202.

In Step 205, the header part of the IP data packet is read from the received ATM data packet. In the following Step 206, counter value for the number ~~plurality~~ of switchings of the IP data packet that have already ensued is reduced by the numerical value one.

In a Step 208, a check is carried out to see whether the counter ~~contour~~ value is equal to zero. When this is the case, then the IP data packet, to which the ATM data packet being processed at the moment belongs, has already been switched too often. In order to avoid an overload of the network 10 (see Figure 1) all ATM data packets of this IP data packet are erased after reception. The measures needed for this purpose are implemented in a Step 210. The method is continued in Step 202 following the Step 210.

When, by contrast, it is found in Step 208 that the counter value has not yet reached the numerical value of zero, then a Step 211 immediately follows the Step 208, the header part of the IP data packet being further-processed in said Step 211 in that, for example, the Internet protocol version number, the data field for the priority or the control data is ~~are~~ interpreted.

In a Step 212, a new connection identifier for the received ATM data packet and, thus, for the IP data packet to which the received ATM data packet belongs is subsequently determined with the assistance of the IP

revaluation memory 102 and the destination address that was already read in Step 205. In a Step 214, the new connection identifier is then entered into the ATM revaluation memory 104. The error checking method prescribed in the Internet protocol is implemented in a Step 216. On the one hand, an intermediate result CRC* is determined for the received ATM data packet according to error checking methods and, on the other hand, an intermediate result CRC is likewise determined according to error checking methods for a new ATM data packet generated from this ATM data packet. For example, the counter value is reduced by the numerical value one in the new ATM data packet. Upon implementation of the error checking method for the new ATM data packet, an intermediate result CRC deviating from the intermediate results CRC* therefore arises. Both intermediate results CRC* and CRC are stored in the memory 108.

In Step 218, the new ATM data packet is stored in the memory 106 until all new ATM data packets of the IP data packet being processed at the moment are present.

In a Step 220, a check is subsequently carried out on the basis of the now known length of the IP data packet to see whether further ATM data packets belong to this IP data packet. When this is not the case, then the method is continued in a Step 232 to be explained later. When, however, even more data packets belong to the IP data packet being processed at the moment, then the method is continued in method Step 222.

In Step 222, a check is carried out with the assistance of the ATM revaluation memory 104 to see whether an ATM cell belongs to the IP data packet being processed at the moment. When this is not the case, a standard processing that corresponds to the standard processing in Step 204 ensues in Step 224.

When, by contrast, it is found in Step 222 that the received ATM data packet belongs to the IP data packet being processed at the moment, then, in a Step 226, the error checking method is continued with the received ATM data packet, whereby a new intermediate result CRC^*_{new} is generated proceeding from the intermediate result CRC^*_{old}. A new intermediate result

CRC_{new} is calculated from the intermediate result CRC_{old} for the ~~payload~~ message data of a new ATM data cell that is generated from the most recently received ATM data cell. The intermediate results CRC^*_{new} and CRC_{new} are again stored in the memory 108. These intermediate results are then the old values CRC^*_{old} and CRC_{old} when the Step 226 is processed next.

A new connection identifier is determined for the new ATM data packet from the ATM revaluation memory 102. A new ATM data packet generated with this connection identifier is then stored in the memory 106 - Step 228.

In a Step 230, a check is subsequently carried out to see whether further ATM data packets belong to the IP data packet being processed at the moment. When this is the case, then the method is continued in Step 222. The method is thus in a loop composed of the method steps 222 through 230. When running this loop, new ATM data packets are successively generated for all received ATM data packets of the IP data packet being processed at the moment. These data packets are all stored in the memory 106. The intermediate results CRC* and CRC of the two error checking methods are also updated in the memory 108, keeping step.

The loop composed of the method steps 222 through 230 is only left in Step 230 when all ATM data packets of the IP data packet being processed at the moment have been processed. When this is the case, the aforementioned Step 232 immediately follows Step 230.

In Step 232, a reference value CRC^*_{ref} is read from the most recent ATM data packet of the IP data packet being currently processed, this being a rated value for the error checking method implemented with the received data packets.

A check is carried out in a Step 234 to see whether a reference value CRC^*_{ref} agrees with the result CRC* calculated for the received ATM data packets of the IP data packet being processed at the moment. When this is not the case, a transmission error must be present, and all new ATM data packets are to be discarded, Step 236. The method is again continued in Step 202 after the Step 236.

When, by contrast, it is found in Step 234 that the intermediate result CRC* calculated for the received ATM data packets agrees with the rated value CRC^*_{ref}, then a Step 238 immediately follows the Step 234. The Step 238 is thus only implemented when no transmission errors occurred in the transmission of the ATM data packets belonging to an IP data packet. The new ATM data packets for this IP data packet stored in the memory 106 (see Figure 3) contain valid data. When generating the most recent, new ATM data packet for the IP data packet being processed at the moment, the old reference value CRC^*_{ref} ~~[sic]~~ is overwritten with the intermediate value CRC calculated for the new ATM data packets.

Subsequently, the new ATM data packets of the IP data packet being processed at the moment are sent - in a Step 240 - from the line unit 20 via the data line 110 to the switching network 44. The Step 240 is again followed by the Step 202. Previously, however, the new connection identifier in ~~th~~ the ATM revaluation memory is also overwritten by the entry "IP".

Even though the method was explained only for an IP data packet being processed at the moment, the ATM data packets for a ~~plurality~~ number of IP data packets can be simultaneously processed. The method explained on the basis of Figures 6A and 6B is thereby implemented for each of these IP data packets. There are then a ~~plurality~~ number of connection identifiers, four, for example, ATM data packets coming from the end system A in the ATM revaluation memory. Instead of a connection identifier VPI_A/VCI_A, a ~~plurality~~ number of connection identifiers VPI_A/VCI_{A1}, VPI_A/VCI_{A2}, etc., are then employed.

LIST OF REFERENCE CHARACTERS

10	Data communication network, network
IP	Internet protocol
ATM	Asynchronous transfer mode
12, 14, 16	Internet computer
18	Transmission line
20	Line unit
22	Switching unit
24	Sub-network
26, 30	Transmission line
28, 32	Line unit
34	Transmission line
36	Line unit
38, 40	Internet computer
42	Controller
44	ATM Switching network
50	AAL5 data frame
52 through 58	ATM data packet
60, 60', 60"	IP data packet
62	Header part
64, 64', 64"	Payload part
66	Filler data
68	Reference data
100	Processing unit
102	IP revaluation memory
104	ATM revaluation memory
106	Memory
108	Memory
CRC, CRC*	Intermediate values
110, 112	Transmission line
120	Internet computer
122	Transmission line

124	Line unit
A, B, C	End system
IP-ADR	Internet address
VPI_A/VCI_A	Connection identifier
VPI_B/VPI_B [sic]	Connection identifier
VPI^*_B/VCI^*_B	Connection identifier
VPI_C/VCI_C	Connection identifier
VPI^*_C/VCI^*_C	Connection identifier
126, 128	Connection
150 through 154	Data field
156	Data block
158, 160	Address field
162, 164	Memory cell in the ATM revaluation memory
166 through 172	Memory cell in the IP revaluation memory
200	Start
202	First ATM cell of a frame?
204	Standard processing
205	Read frame header
206	Reduce switching counter
208	Switching counter = 0?
210	Erase ATM cells of the frame
211	Further-process frame header
212	Revaluate the IP address
214	Update ATM revaluation memory
216	Determine CRC*, CRC
218	Store processed ATM cell
220	Does frame have further ATM cells?
222	Does ATM cell belong to the frame?
224	Standard processing
226	$CRC^*_{old} \rightarrow CRC^*_{new}$; $CRC_{old} \rightarrow CRC_{new}$
228	Store processed ATM cell
230	Does frame have further ATM cells?

232	Read CRC^*_{ref} from last ATM cell
234	$CRC^* = CRC^*_{ref}$?
236	Erase ATM cells of the frame
238	Overwrite CRC^*_{ref} with CRC
240	Send ATM cells of the frame